Distribution. A; Public Release
This work is sponsored by the Department of Defense under Air Force contract FA8702-15-D-0001. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Finding Malicious Cyber Discussions in Social Media

Richard P. Lippmann, William M. Campbell, David J. Weller-Fahy, Alyssa C. Mensch, Giselle M. Zeno, and Joseph P. Campbell

Today's analysts manually examine social media networks to find discussions concerning planned cyberattacks, attacker techniques and tools, and potential victims. Applying modern machine learning approaches, Lincoln Laboratory has demonstrated the ability to automatically filter cyber discussions from Stack Exchange, Reddit, and Twitter posts written in English.

Criminal hackers often use social media networks to discuss cyberattacks, share strategies and tools, and identify potential victims for targeted attacks. Analysts examining these discussions can forward information about malicious activity to system administrators who can then detect, defend against, and recover from future attacks (Figure 1). For example, prior to the anticipated cyberattacks on Israeli government websites by the hacking group Anonymous, government analysts were monitoring hackers on Facebook and in private chat rooms. As a result, system administrators were prepared to counter distributed denial-of-service attacks and defacement of government websites. Israel temporarily suspended some international traffic to these sites and advised employees to not open emails for five days. Teams were available to respond to successful attacks and repair or restore websites. Because of Israel's careful preparation, this cyberassault only succeeded in bringing down a few websites for a short period of time [1].

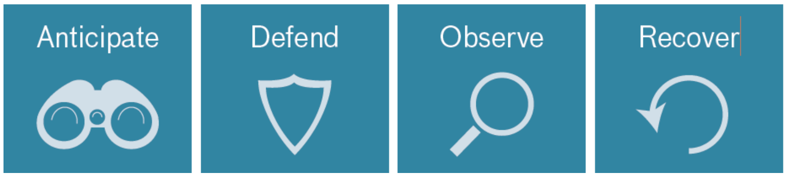


FIGURE 1. The four components pictured above must be present in any security process. Anticipating an attack enhances the defense of systems, observation of attacks and responses, and recovery efforts.

Monitoring social media networks is a valuable method for discovering malicious cyber discussions, but analysts currently lack the automation capabilities needed to sift through vast amounts of data. Analysts try to discover and track cyber discussions by manual searches, often using metadata (e.g., account names, thread or discussion topics, sources and destinations of social media discussions). This process is labor intensive, particularly when non-English cyber discussions must be manually translated, and sometimes ineffective because attackers can easily change metadata to hide malicious conversations by adopting innocuous-sounding names for Stack Exchange topics, Reddit threads, or Twitter hashtags. A more efficient and effective method is to supplement metadata analysis with direct mining of the discussion text via machine learning and human language technology (HLT) approaches. Such approaches can be applied to English and non-English content without requiring manual translation.

Although great bodies of published work focus on either HLT or cybersecurity, surprisingly few publications discuss the application of HLT to the cyber domain. The application appears to have been first proposed by Klavans in 2013 [2]. More recently, Lau et al. analyzed interactions between known cybercriminals on social media to distinguish between *transactional* interactions, in which cyberattack tools are bought or sold, and *collaborative* interactions, in which cybercriminals share tools or information without any monetary exchange [3]. However, their analysis requires manual extraction of cyber discussions before automated transaction analysis can be performed.

An Automated Solution (*heading level 1*)

Under the Cyber HLT Analysis, Reasoning, and Inference for Online Threats program, Lincoln Laboratory is developing HLT classifiers to automatically detect cyber discussions concerning attack methods, defense strategies, and tools' effectiveness through the examination of online forums. Our aim is to leverage available techniques, such as topic classification, entity recognition, and sentiment analysis (i.e., opinion mining), which have only begun to be applied to the problem of detecting and analyzing malicious cyber discussions.

Concept of Operations (*heading level 2*)

Among the large number of online discussions, few are on cyber topics. Our goal is to utilize modern HLT approaches to automatically filter out those cyber discussions for analysts (Figure 2).

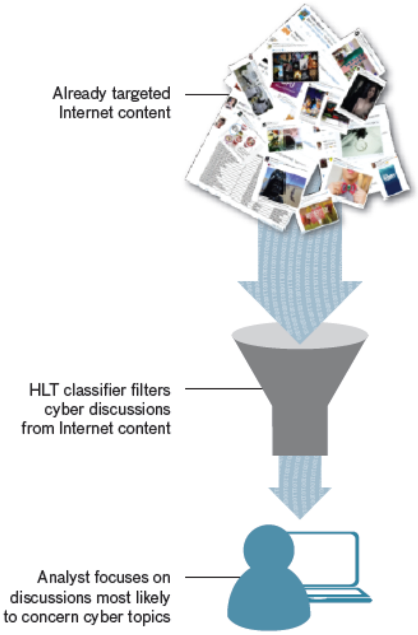

Already targeted Internet content
HLT classifier filters cyber discussions from Internet content
Analyst focuses on discussions most likely to concern cyber topics

FIGURE 2. An automated process for extracting cyber discussions from online forums reduces the amount of time an analyst needs to spend on eliminating content that is irrelevant to his or her investigation.

We identified two concepts of operations (CONOPS) for using an HLT machine learning classifier to determine if a discussion concerns malicious cyber topics:

1. An analyst has already discovered Internet content, such as lists of topics in Reddit or lists of users in Twitter, to examine. Instead of an analyst manually examining all discussions grouped under these topics or all tweets posted by these users, a classifier trained to determine whether a discussion/tweet was about cyber topics could identify which content an analyst should focus on first. This ranking is necessary because discussions may drift from topics of interest (malicious cyber topics) to topics that are not of interest (nonmalicious cyber topics and noncyber topics) and vice versa, or they may move to users who do not discuss malicious cyber topics.
2. An analyst is trying to discover Internet forums (e.g., Stack Exchange communities) that contain cyber discussions of interest. This scenario is more difficult—the search is not focused on known forums and is thus wider. When exploring new Internet discussion areas, the classifier can rank the forums by their probability of containing cyber content, prioritizing discussions for an analyst's investigation. For best performance, the classifier should be trained to find new discussions that are similar to past ones of interest.

Classifier Development (*heading level 1*)

Before an HLT classifier can filter out cyber discussions, it must first be trained on cyber and noncyber discussions. In the sections below, we describe how training and testing were performed for our HLT classifiers. We also describe how data were gathered and labeled to support classifier development and how a previously developed keyword classifier was used as a reference for performance evaluations.

Training (*heading level* 2)

The first training phase required to create an HLT classifier involves selecting both cyber and noncyber social media discussions to be fed into the classifier. To ensure that highly ranked discussions are actually the discussions of most interest to analysts, cyber examples used for training should be representative of those that were of most interest in the past. Training data should contain noncyber discussions that cover many topics and should capture words and phrases that distinguish cyber from noncyber content in many subjects to prepare the classifier for the diversity of content it will encounter once operational.

After an HLT classifier is trained, it can be fed input text from a discussion occurring on a social media network and provide as output the probability that the discussion is on a cyber topic (Figure 3). An output probability supports both CONOPS: conversations in forums of interest can be ranked by probability and analysts can examine those with the highest probabilities first, or many new forums can be scanned to identify those with the greatest number of high-probability cyber conversations.

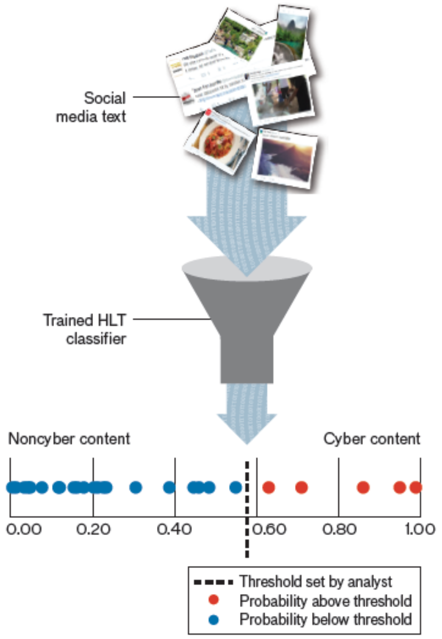


FIGURE 3. Text from a social media discussion is fed into a trained human language technology (HLT) classifier. The classifier then outputs a probability, based on a threshold set by an analyst (dashed line), that the discussion is about cyber topics. This probability ranges from zero (not about cyber) to one (almost certainly about cyber). Output probabilities for different discussions are shown above. An analyst would examine all discussions with probabilities above the threshold (red dots) and ignore remaining discussions with probabilities below the threshold (blue dots).

Social Media Corpora (*heading level 2*)

Initially, we are training and testing our classifiers using three social media networks that analysts may monitor: Stack Exchange, Reddit, and Twitter (Table 1). Stack Exchange is a well-moderated question-and-answer network with communities dedicated to many diverse topics. Answers can be quite comprehensive, long, and well written. Reddit is a minimally moderated set of forums with main topics called sub-Reddits and many individual threads or discussions under each topic. Twitter data consist of short tweets with at most 140 characters each. Tweets can be followed via usernames, hashtags that identify tweets on a similar topic, or Twitter lists (i.e., curated groups of Twitter users).

These three corpora were selected because they

- contain text with at least some cyber content;
- span a range of social media types; and
- offer a history of prior posts over a long time span.

For each of these corpora, posts and comments in those posts were gathered to generate cyber and noncyber “documents” to be fed into our classifiers for training and testing.

Table 1. Characteristics of Social Media Posts

Social Media Corpus	Post Characteristics	Example Post
StackExchange	Long, curated posts	“Every time I try even a simple stack smash on a 64bit machine, I run into issues. An address I am trying to write always contains null bytes.”
reddit	Medium-length, not-well-curated posts	“What is a hack that you know that is awesome or mind blowing?”
twitter	Short (140 characters), noncurated posts	“Cyber attack creates temporary disruption in Hawaii’s thirty-meter telescope website http://bit.ly/1OXOdce #cybersecurity #infosec”

Document Labeling (*heading level 3*)

Documents refer to a collection of all posts concerning discussions on a specific question for Stack Exchange, all posts for a specific sub-Reddit thread in Reddit, and all collected tweets from a specific Twitter user (with more than 20 tweets but less than 300 tweets; this range was selected to create a balanced set of training data, as Twitter users, particularly spammers, may have 1000s or 10,000s of tweets).

Preprocessing eliminated dates, thread titles, hashtags, usernames, and other metadata so that the classifier would be trained using only the discussion text (when a trained classifier is put into operational use, metadata may not be available to provide context for a discussion). Documents for Stack Exchange and Reddit were labeled with topic titles and tags set by the users of each corpus. All posts under cyber-related topics (e.g., *reverse engineering, security, malware, blackhat*) were labeled as cyber, and posts on other topics (e.g., *astronomy, electronics, beer, biology, music, movies, fitness*) were labeled as noncyber. For Stack Exchange, we further restricted cyber discussions to posts with lower-level tags (e.g., *penetration test, buffer overflow, denial of service, Heartbleed*[1]). For Twitter, tweets from 127 users identified as cyber experts by Lincoln Laboratory researchers were labeled cyber, while tweets from 500 other randomly selected users were labeled noncyber. Table 2 shows for each corpus the number of cyber and noncyber topics, the number of documents, the median number of words in each document, the time period covered by the collection, and a summary of how documents were labeled as cyber or noncyber.

Table 2. Social Media Corpora Document Labeling

	TOPICS		DOCUMENTS			
CORPUS	CYBER	NONCYBER	NUMBER OF DOCUMENTS	MEDIAN NUMBER OF WORDS	TIME COVERED	DOCUMENT LABELING METHOD
Stack Exchange	5	10	~200K	245	Years	Cyber-related topics and tags
Reddit	10	51	~59K	152	Months	Cyber-related sub-Reddits
Twitter	127	500	627	546	Months	Expert cyber users’ tweets

[1] Made public in April 2014, Heartbleed is a vulnerability in the OpenSSL cryptography library that allowed attackers to steal servers’ private keys and users’ passwords.

Reference Keyword Detector (*heading level 2*)

To compare the performance of our classifier with that of previously used classifiers, we implemented a tool that detects cyber discussions via keywords and phrases. It searches for 200 cyber keywords and phrases in a document, counts the number of occurrences, and normalizes the count by dividing the total number of occurrences by the total number of words in the document. Higher counts indicate documents that are more likely about cyber topics. Cyber discussion keywords (e.g., *rootkit, infected, checksum)* and phrases (e.g., *buffer overflow, privilege escalation, distributed denial of service)* were selected by trained linguists.

Processing and Classification (*heading level 2*)

As shown in Figure 4, the classification pipeline requires preprocessing each document, generating features for each word in each document, and training a classifier to distinguish between cyber and noncyber documents on the basis of these generated features. The preprocessing step employs stemming[2] to normalize word endings and text normalization techniques, such as the removal of words containing numbers and the replacement of URLs with a token indicating a URL was used, to ensure that the feature inputs are standardized. When generating features, we used term frequency–inverse document frequency (TF-IDF) values, created by counting the occurrences of words in documents and the number of documents in which the words occur. In our research, and in the HLT community's research in general, TF-IDF-based features have provided good performance when used in text classification. Our experiments used the TF-IDF values of unigrams (individual words) to create features. To classify the documents on the basis of these features, logistic regression and linear support vector machine classifiers were used; both classifiers train rapidly, require little computation to analyze a document, and provide an output score proportional to the probability that the input document contains cyber content.

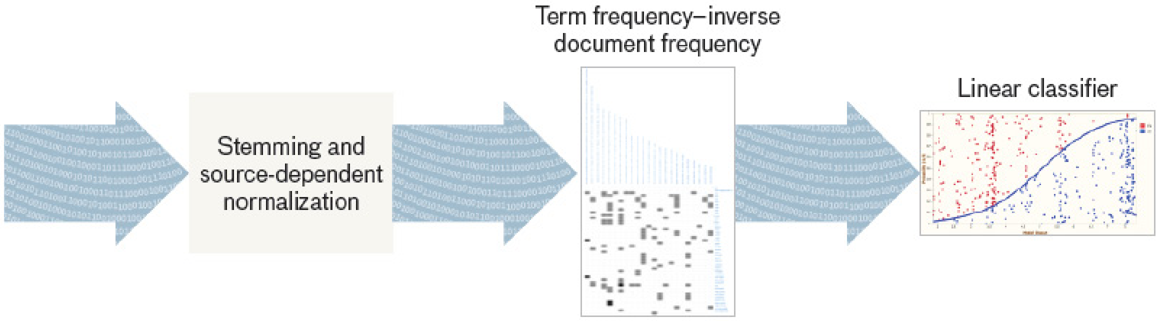

Term frequency–inverse document frequency
Stemming and source-dependent normalization
Linear classifier

FIGURE 4. The flow of documents through the classification pipeline requires preprocessing to ensure the text is ready to use in feature generation, calculation of term frequency–inverse document frequency values for each word in the document, and classifier training using the features generated for each document.

Initial Results (*heading level 1*)

Figure 5 shows initial results for classifiers trained and tested on Stack Exchange, Twitter, and Reddit data. Each classifier outputs the probability that each document discusses cyber topics; this probability is based on a set threshold (the minimum probability required for the classifier to label a document as cyber). The document labels then make it possible to determine the number of false alarms (i.e., noncyber documents that are classified as cyber) and misses (i.e., cyber documents that are classified as noncyber).

[2] Stemming is the reduction of a word to its root form, e.g., stemming "hacks" or "hacked" produces "hack."

We present our results in the form of detection error tradeoff (DET) curves that show how false-alarm and miss probabilities vary as the threshold on the classifier's output probability varies as plotted on normal deviate scales [4]. Our goal is to provide good detection of cyber documents (e.g., a low miss rate) and limit the number of noncyber documents that are labeled as cyber (i.e., a low false-alarm rate). As shown by the gray box in Figure 5, a false-alarm rate below 1% and a miss rate below 10% is the performance target. Within this target range, our pipeline provides good filtering of Internet content as long as the portion of cyber documents relative to all documents presented to a classifier is 5% or greater.

The curves shown in Figure 5 indicate that the classifiers we developed for each social media corpus do meet the performance target—they miss less than 10% of cyber-labeled documents and classify less than 1% of the noncyber-labeled documents as cyber. Before obtaining these results, we first had to understand the minimum number of words in each document, amount of training data, and types of preprocessing necessary to provide good performance.

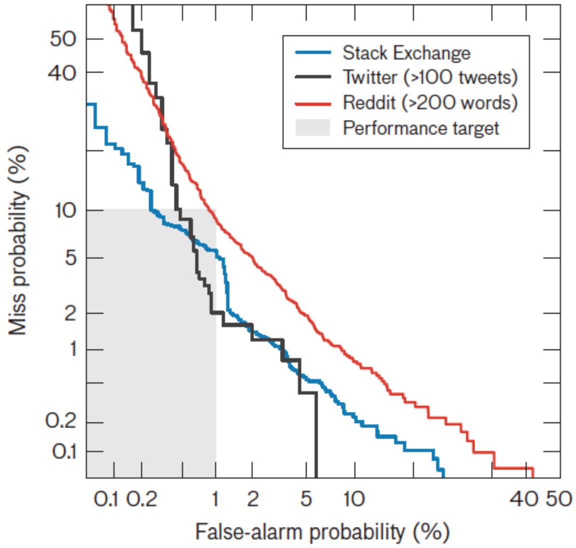


FIGURE 5. Initial detection error tradeoff results indicate that the classifiers perform well for all three social media corpora; all curves overlap with the performance target region (gray box).

Comparative Analysis of Classifiers (*heading level 2*)

Figure 6 compares the performance of the baseline keyword classifier to the logistic regression classifier on Stack Exchange data. The logistic regression classifier (blue curve) passes through the performance target region, meaning it misses less than 10% of cyber documents with a false-alarm rate of less than 1%. The baseline keyword system (black curve) performs substantially worse than the logistic regression classifier. At a false-alarm probability of 10%, the system fails to detect roughly 40% of the cyber documents; at a false-alarm probability of 1%, the miss probability is roughly 60%. To determine the cause of this poor performance, we examined the Stack Exchange documents that corresponded with the false alarms. We found that false alarms were often caused by one or more occurrences of cyber keywords in documents with topics unrelated to cyber. For example, the keyword *infected* appeared in documents referring to bacterial infection. Similarly, the keyword *checksum* appeared in many documents on technical topics. Simply counting occurrences of keywords without considering the context of the documents led to the false alarms. Worst-case performance, shown by the chance-guessing curve (red), is obtained by randomly assigning a label to each document.

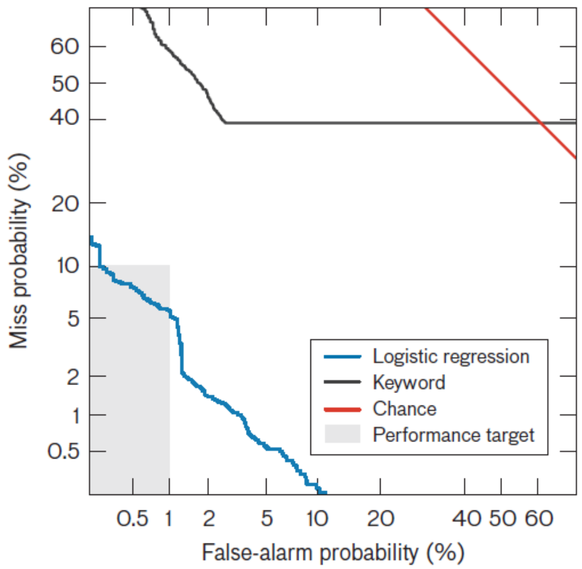


FIGURE 6. The detection error tradeoff curves for Stack Exchange documents show that the logistic regression classifier significantly outperforms both the baseline keyword system and chance guessing.

Table 3 provides some insight into why our logistic regression classifier performs better than the keyword system. On the left are the 50 words that receive the highest positive weights (i.e., the words that are most useful to our classifier in identifying cyber documents) and thus contribute more than other words in causing a document to be classified as cyber. These words span a wide range of cyber discussions on several topics. Many of these words and other positively weighted cyber words used by this classifier are highly likely to be present in cyber documents. While there is some vocabulary drift with time, experiments suggest that most terms remain stable for up to one year (see section titled "Stability in Performance over Time"). Unlike the keyword system, our classifier strongly indicates cyber only if many of the 50 cyber words are combined in one document. Multiple instances of one word will not yield a strong cyber indication. The right side of this table lists the 50 words that receive the highest magnitude negative weights (i.e., the words that are most useful to our classifier in identifying noncyber documents) and thus contribute more than others in causing a document to be classified as noncyber. These words indicate the breadth of topics that noncyber documents cover. This diversity suggests that a large set of noncyber documents needs to be fed into the logistic regression classifier during training.

Table 3. List of Most Important Cyber and Noncyber Words Used by Our Logistic Regression Classifier Trained on Stack Exchange Data

Top 50 Cyber Words	Top 50 Noncyber Words

HTTP, SQL, Secur, URL, Window, access, address, app, application, attack, authenticate, browser, bug, certificate, client, code, crack, detect, encrypt, execute, exploit, file, firewall, hash, infect, inject, install, key, malicious, malware, network, obfuscate, overflow, packet, password, payload, request, risk, scan, script, secure, server, site, test, tool, traffic, user, virus, vulnerability, web	*Arduino, Christian, God, LED, The, and, bank, board, buy, cell, chip, chord, circuit, clock, credit, current, datasheet, design, electron, film, frac, frequency, fund, graph, hi, invest, microcontroller, motor, movie, music, note, output, part, pin, play, power, rate, resistor, serial, signal, simulate, state, stock, tax, the, time, tree, two, voltage, wire*

The Effect of Document Length and Amount of Training Data (*heading level 2*)

The DET curves in Figure 7 show how our classifier's performance depends on the number of words in a Reddit document. For comparison purposes, the left plot shows how poorly the classifier performs when all documents (no minimum word count, many short threads with no responses) are included (black curve). The right plot shows the classifier performance with minimum word counts in smaller increments, allowing a better view of the performance improvements. As seen in both plots, performance initially increases rapidly as the number of words increases. However, the rate of performance increase slows as the minimum number of words increases, and classifier performance enters the target range when the minimum number of words is above 200. Our results thus suggest that 200 or more words in an Internet conversation are required to provide accurate classification of cyber and noncyber documents. To examine the effect of the amount of noncyber Reddit data on performance, the number of noncyber topics was increased from 10 to 51 (Figure 8). A small performance improvement is seen for this increase in the number of noncyber topics.

Classifier performance also improves for Twitter as the number of words per document and the amount of noncyber training data are increased while the number of cyber users (127) remains constant (Figure 9). For Twitter, a document is composed of all the tweets from a single user, so the number of words per document is increased by including more tweets per user. The number of noncyber training documents is increased by randomly sampling users and collecting their tweets in additional documents. Because we assume that there is a very low probability of a randomly sampled user discussing cyber topics, no extra labeling or cost is incurred by incorporating additional training data. On average, there are 10 words per tweet after preprocessing, so in each of the results with a minimum of 20 tweets, there are 200 words per document. Performance was further improved by collecting additional tweets and increasing the average number of words per document to 1000. These results are consistent with the Reddit results showing improved classifier performance as more words are added to the documents and with the Reddit and Stack Exchange results showing improved classifier performance as more noncyber training data are provided.

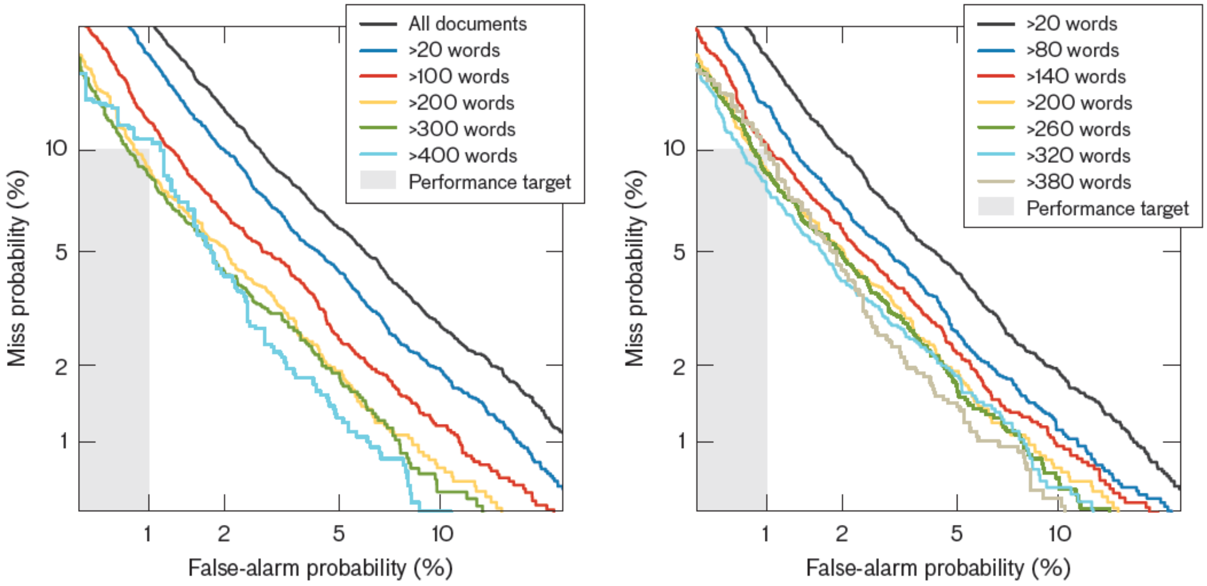


FIGURE 7. As the minimum number of words in each Reddit document is increased, the classifier's performance improves.

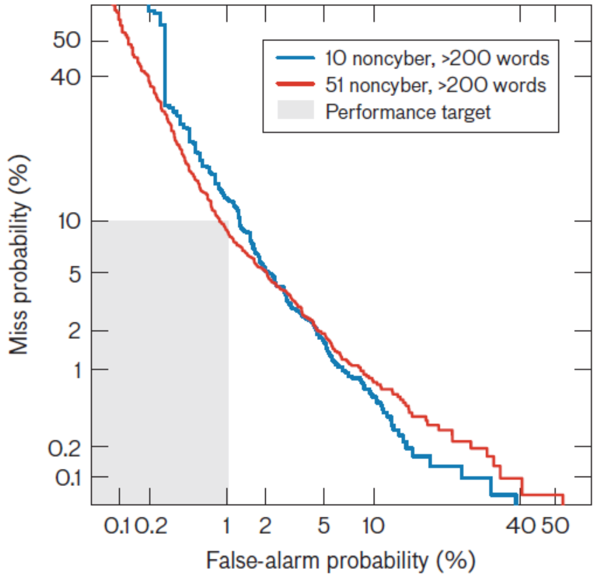


FIGURE 8. Increasing the number of noncyber sub-Reddit topics from 10 to 51 reduces the gap to the performance target. Both experimental runs were performed with a minimum word count of 200. Note that the performance increases by approximately 3% in the desired false-alarm range.

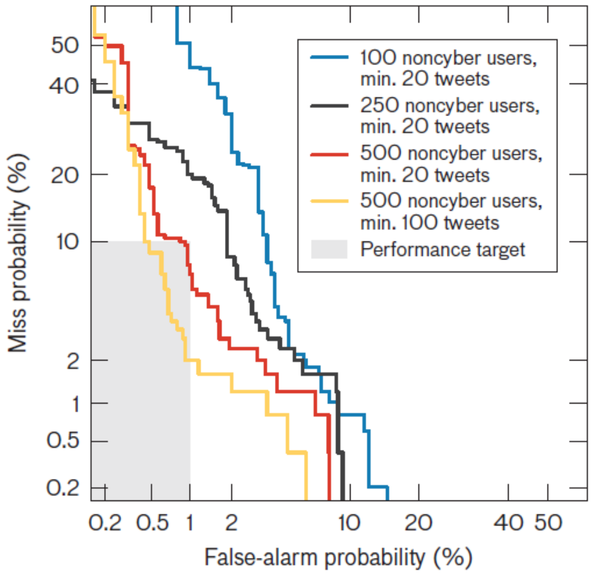


FIGURE 9. As the number of noncyber Twitter users and words per document (i.e., tweets per user) is increased, performance improves. For example, at 1% false alarms, the miss rate is 40% with 100 noncyber users (blue curve), 20% with 250 users (black curve), and only 6% with 500 users (red curve). By adding additional tweets from the 500 users, the miss rate is reduced to 2% at 1% false alarms (yellow curve).

Stability in Performance over Time (*heading level 2*)

Another test of our logistic regression approach determined whether a classifier trained before the Heartbleed vulnerability was made public could detect social media discussions concerning Heartbleed. Such discussions could only be found if they included words that were used in prior social network cyber discussions because the classifier would have never seen the word *Heartbleed*. Figure 10 plots the cumulative percentage of Stack Exchange threads detected by a logistic regression classifier trained on 3924 cyber and 7848 noncyber documents posted before the Heartbleed attack was announced on 8 April 2014. The classifier immediately detects the flurry of posts on 8 April and in the following days. Of the 106 *Heartbleed*-tagged threads, 86% were detected and only 14% were missed at a false-alarm rate of 1%. Our logistic regression classifier performed much better than the keyword baseline system, which only detected 5% of the Heartbleed discussions, because ours detects words related to the protocols affected by Heartbleed (e.g., *SSL*, *TLS*) and other words associated with cyber vulnerabilities (e.g., *malware*, *overflow*, *attack*). Because the keyword system lacked such keywords used in Heartbleed discussions, it suffered from a high miss rate.

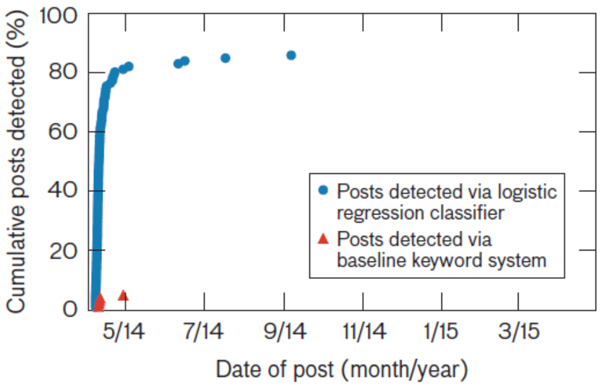


FIGURE 10. Our logistic regression classifier, which was trained on data before the Heartbleed vulnerability was known, was still able to detect 86% of Stack Exchange posts discussing Heartbleed (blue dots). By comparison, the baseline keyword system only detected 5% of posts discussing Heartbleed (red triangles).

A system to detect cyber documents is most useful if it does not require frequent retraining to match possible changes in cyber vocabulary over time. We performed experiments in which a classifier was trained on Stack Exchange data up to a given date and then tested every month after that date without retraining. Figure 11 plots the miss percentage (averaged over false-alarm rates ranging from 0.25% to 1.0%) for a classifier that was trained on data before June 2012 and then tested each month for a year on new data appearing within each respective month. The results indicate that the miss rate increases little over the year and is always below roughly 10%. The experiment was repeated over multiple time periods from 2012 through 2014, producing similar results each time. Classifiers thus do not require frequent retraining—once a year or at most every six months is adequate.

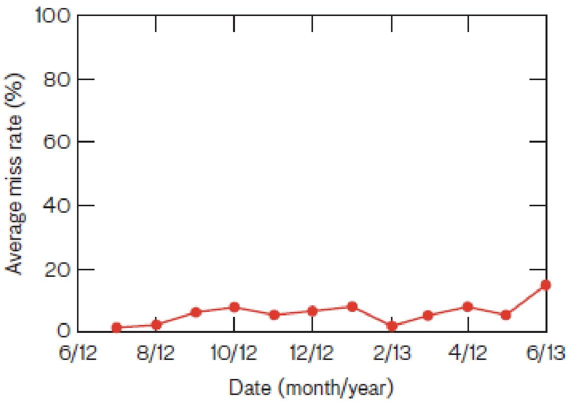


FIGURE 11. A logistic regression classifier was trained on Stack Exchange data before June 2012 and then tested every month after that for a year on new data. Despite the classifier not being retrained, its miss percentage increased little over the year and stayed within the performance target of below 10%.

Filtering and Concentrating Cyber Documents *(heading level 2)*

One of our goals with the cyber classifiers we are developing is to have them filter or concentrate documents from social media sources so an analyst is presented mainly with cyber documents. We assume that our classifiers will be applied to preselected Internet data that are known to have more than 1% cyber documents and that a 90% detection rate for cyber documents is sufficient to discover important long-standing cyber discussions. As previously discussed, the target performance we have been using as a reference is a miss percentage below 10% for a false-alarm percentage below 1%. Figure 12 shows the filtering or concentration effectiveness of our classifiers with performance in this target range when the classifiers are applied to Internet sources with different initial concentrations of cyber documents. The vertical axis in this figure is the fraction of cyber documents remaining after filtering the documents; the horizontal axis is the fraction of cyber documents in the Internet source. The upper curve (red) is for a classifier that misses 10% of the cyber documents with 0.25% false alarms and the lower curve (blue) is for a classifier that misses 10% of the cyber documents with 1% false alarms. If only 1 in 100 of the Internet documents examined are cyber (1% on the horizontal axis), then our classifiers that provide performance between these curves present between 50% (1 in 2) and 80% (4 in 5) cyber documents to an analyst. This ability to enrich output of cyber documents is a large improvement in concentration over the existing keyword classifier, which presents 30% (3 in 10) cyber documents to an analyst at a 1% false-alarm rate. If the fraction of cyber documents increases to only 5% (1 in 20), our classifiers present between 83% (5 in 6) and 95% (19 in 20) cyber documents to an analyst. These results motivate the performance target we are reaching with our classifiers and suggest that our classifiers are useful even if there is only one cyber document in each 100 documents from an Internet source.

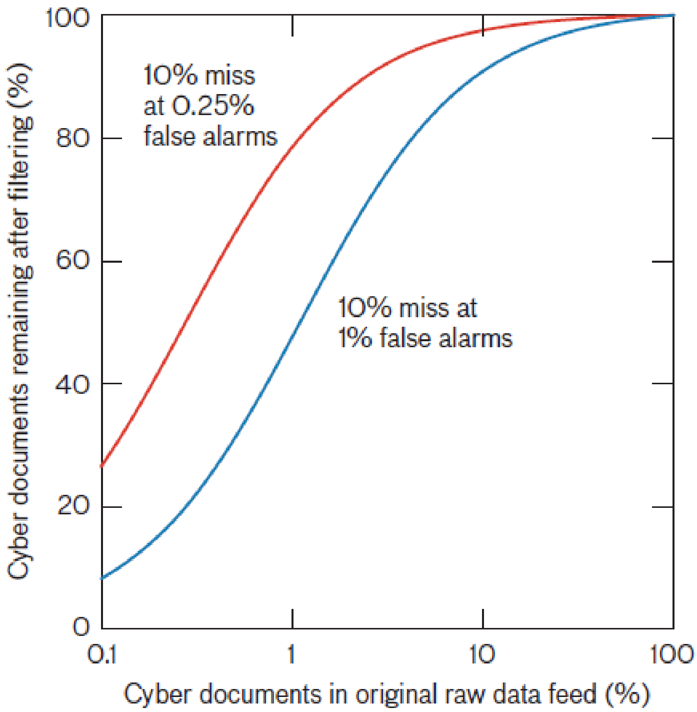


FIGURE 12. The two performance curves for our classification pipeline show the percentage of cyber documents that would be presented to an analyst after classification (*y*-axis) compared to the percentage of cyber documents in the input documents before classification (*x*-axis).

Related Work (*heading level 1*)

Relational Classification Methods (*heading level* 2)

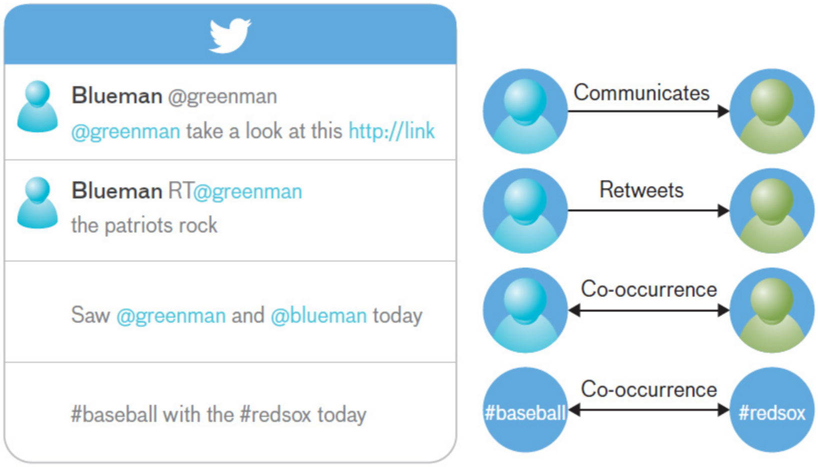
Up to this point, we have focused on extracting the language content within social media posts to perform classification. Certain social media networks, such as Twitter, include rich metadata (e.g., user, content, messaging information) that can be leveraged to build a social network of entities describing the relations and activities between these entities [5]. Entity types may include groups, individuals, and even hashtags. Because of homophily ("birds of a feather flock together"), we expect that finding one cyber user on

Twitter will lead to finding other cyber users who follow or retweet each other. Homophily is part of a more sophisticated set of relational classification methods [6] that combine social network metadata and machine learning techniques to establish connections and interactions among users and content on the network.

The steps for relational cyber classification are as follows: First, text and metadata of a single message are processed to produce entities and the relations between them [5]. For example, a tweet by @cyberuser, such as "@cyber01 Look at this #malware exploit," shows a relation between the two Twitter users, @cyberuser and @cyber01. It also shows a relation between the two users and the hashtag #malware. Second, the entities and relations are combined in a database that stores graphs and optimizes graph operations (i.e., a graph database), such as finding all the neighbors of a node (an entity). Computed graph features, such as the number of nodes connected to a given fixed node, can be added to the graph along with attributes on relations and entities (e.g., full names, email addresses). The final step is to apply relational learning to the problem of classifying entities as cyber/noncyber, a process that consists of finding relational features for both entities and related entities; then, labels of nodes representing known cyber users and homophily are used to boost performance of classifying nodes as cyber or noncyber. This relational learning technique is referred to as collective classification or semisupervised learning in the literature [7–9]).

Information Extraction and Graph Construction (*heading level 3*)

The first two steps, information extraction and graph construction, are performed by using multitype nodes and edges (relations between entities). Figure 13 shows the basic process, with four different types of relations and two types of entities being used to construct a Twitter graph.



Blueman @greenman
@greenman take a look at this http://link

Blueman RT@greenman
the patriots rock

Saw @greenman and @blueman today

#baseball with the #redsox today

FIGURE 13. A Twitter graph is constructed by using multiple edge types (communications between users, retweets, co-occurrence of users, and co-occurrence of hashtags) and two types of nodes: users (@greenman, @blueman) and hashtags (#baseball, #redsox).

Relations and entities capture a significant amount of the activity on Twitter. Applying the method described in Figure 13 on 10% of the tweets posted for a typical month on Twitter in 2014 yields a graph with the following characteristics:

- 52.3 million nodes (6.7 million hashtags and 45.6 million users)
- 361.7 million edges

This large graph can be stored in a graph database (e.g., Neo4j) and explored using graph queries. A typical example of querying for the user "@lennyzeltser" and all of his neighbors in the graph is

```
match (n:user {name:'@lennyzeltser'})-[r:rel]-(m) return n,r,m;.
```

This query yields the result shown in Figure 14. In the center of the graph is the user we queried. Hashtag neighbors (green circles) are #mac4n6 (Mac Forensics), #dfir (Digital Forensics and Incident Response Summit), and #remnux (A Linux Toolkit for Reverse Engineering and Analyzing Malware)—all cyber forensics–related hashtags. Many of the user neighbors (blue circles) are also cyber related (e.g., @malwaremustdie, @malwarejake, @sansforensics), but some are more generally named, for example, @closedanger. This network of neighbors of @lennyzeltser shows the power of relational homophily—neighbors of a cyber user have a strong tendency to also be cyber users.

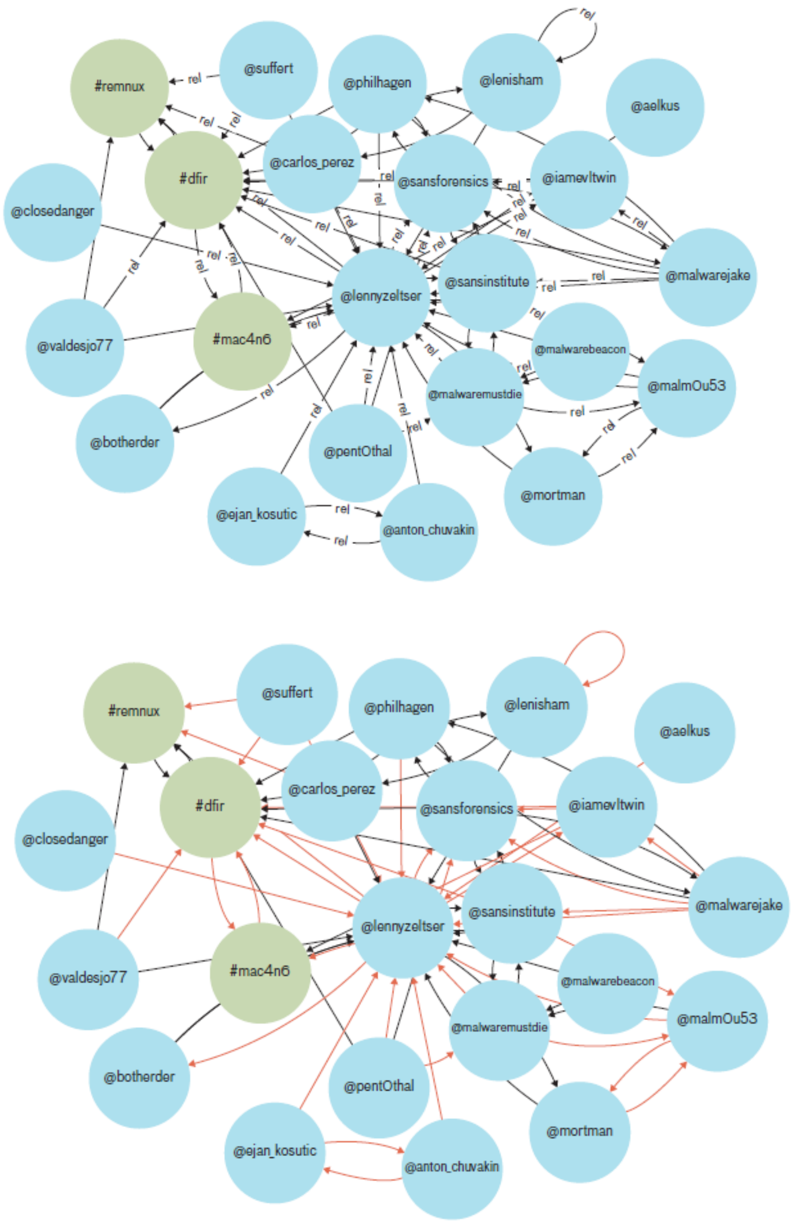


FIGURE 14. An example query in the graph database Neo4j of the user "@lennyzeltser" and of all his neighbors shows how relational homophily can be used to find other cyber users.

After constructing the Twitter graph, we can then utilize relational methods for classification. A standard baseline for relational classification is collective inference [7], which uses the cyber/noncyber probability of a user node and that of its neighbors to iteratively estimate the probability of a user being cyber or not cyber. Thus, collective inference is a natural algorithmic implementation of relational homophily in social networks. Some well-known methods for collective inference are relaxation iteration, Gibbs sampling, iterative classification, and relational dependency networks [7, 10]. Exploring these methods will be an area of future experimentation at Lincoln Laboratory.

Future Work (*heading level 1*)

Our results demonstrate that

- our HLT classifiers performed well for all corpora;
- roughly 200 words in a discussion provide good detection of cyber conversations;
- a classifier trained before the major Heartbleed vulnerability was announced could accurately detect discussions relating to this vulnerability; and
- performance of a classifier is maintained even when tested on discussions occurring six months to a year after it was trained.

However, preliminary experiments suggest that performance degrades when a classifier is trained on one corpus (e.g., Reddit) and tested on another (e.g., Stack Exchange). We are currently exploring three approaches to improve cross-domain performance: (1) constructing a generative probabilistic model of cyber documents that can be used to determine if a new document has a high probability of being cyber without referencing noncyber data; (2) using neural network word embeddings to take advantage of the syntactic and semantic relationships between words; and (3) using features derived from graph analysis of social networks. Feature selection, phrase selection, *n*-gram analysis (i.e., considering words that occur together in documents), and cross-domain training and adaptation will be further explored.

We have also begun collecting non-English social media content to test our approaches with other languages. Future relational-learning experiments using social network structure to perform cyber classification are expected to yield information that should be useful to improve content-based methods of classification (such as the TF-IDF and logistic regression methods discussed in this article). Analysts could leverage relational learning to explore the neighbors of a user in a prioritized manner, investigating closely related users, organizations, events, and topics. Follow-on work also includes efforts to automatically extract entities and relationships and to model cyber threats. This automated extraction and modeling will enable us to categorize documents according to the "Diamond Model" of intrusion analysis (so named for how the model organizes the basic aspects of malicious activity in the shape of a diamond) to assess the capabilities, available infrastructure, and victims of cyber adversaries so we can understand how to observe, understand, and defend against them [11].

Acknowledgments

We are grateful to Julie Dhanraj, Beth Richards, Jason Duncan, Peter Laird, and their subject-matter experts for their support. Vineet Mehta's initial contributions to the Cyber HLT Analysis, Reasoning, and Inference for Online Threats (CHARIOT) program are appreciatively acknowledged. Thank you to Clifford Weinstein, Kevin Carter, and Kara Greenfield for initiating the program and providing helpful insights.

References

1. J. Chartaff, "Announced Cyber Attack on Israel Fizzled," *Homeland Security Today*, 15 April 2014, available at http://www.hstoday.us/briefings/daily-news-analysis/single-article/announced-cyber-attack-on-israel-fizzled/a34d782076af7d0053ded523054619c9.html.
2. J.L. Klavans, "White Paper: Cybersecurity – What's Language got to do with it?" 2013.
3. R. Lau, Y. Xia, and Y. Ye, "A Probabilistic Generative Model for Mining Cybercriminal Networks from Online Social Media," *IEEE Computational Intelligence Magazine*, vol. 9, no. 1, 2014, pp. 31–43.
4. A.F. Martin, G.R. Doddington, T.M. Kamm, M.L. Ordowski, and M.A. Przybocki, "The DET Curve in Assessment on Detection Task Performance," *Proceedings of Eurospeech*, vol. 4, 1997, pp. 1899–1903.
5. W.M. Campbell, E. Baseman, and K. Greenfield, "Content + Context Networks for User Classification in Twitter," Neural Information Processing Systems (NIPS) 2014 Workshop, available at snap.stanford.edu/networks2013/papers/netnips2013_submission_3.pdf.
6. L. Getoor and B. Taskar, eds. *Introduction to Statistical Relational Learning*. Cambridge, Mass.: MIT Press, 2007.
7. S.A. Macskassy and F. Provost, "Classification in Networked Data: A Toolkit and a Univariate Case Study," *Journal of Machine Learning Research*, vol. 8, 2007, pp. 935–983.
8. M. Szummer and T. Jaakkola, "Partially Labeled Classification with Random Walks," in *Advances in Neural Information Processing Systems 14*, T.G. Dietterich, S. Becker, and Z. Ghahramani, eds. Cambridge, Mass.: MIT Press, 2001.
9. X. Zhu, "Semi-Supervised Learning Literature Survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1530, 2007.
10. J. Neville and D. Jensen, "Collective Classification with Relational Dependency Networks," *Proceedings of the Second International Workshop on Multi-Relational Data Mining*, 2003, pp. 77–91.
11. S. Caltagirone, A. Pendergast, and C. Betz, "Diamond Model of Intrusion Analysis," Center for Cyber Threat Intelligence and Threat Research, Hanover, Md., Technical Report ADA586960, 2013, available at http://www.threatconnect.com/files/uploaded_files/The_Diamond_Model_of_Intrusion_Analysis.pdf.

About the Authors

Richard P. Lippmann is a Lincoln Laboratory Fellow working in the Cyber Analytics and Decision Systems Group. He joined Lincoln Laboratory in 2001. His research interests include aids for the hearing impaired, speech recognition, pattern classification, neural networks, and cybersecurity. He has taught three courses on machine learning; has been an IEEE Distinguished Lecturer; won the first *IEEE Signal Processing Magazine* Best Paper Award for an early article on neural networks; and has authored or coauthored more than 100 papers on topics including speech recognition, machine learning, and cybersecurity. He served as the program chair of the Research in Attacks, Intrusions, and Defenses workshop; the Neural Information Processing Systems (NIPS) annual conference; and the NIPS workshop on Machine Learning in Adversarial Environments. He has participated in four national-level government studies on cybersecurity. He received a bachelor's degree in electrical engineering from the

Polytechnic Institute of Brooklyn in 1970 and a doctoral degree in electrical engineering from MIT in 1978.

William M. Campbell is a senior technical staff member of the Laboratory's Human Language Technology Group. He provides leadership and technical contributions in the areas of speech processing, machine learning, and social networks. His speech processing work has resulted in advances in speaker and language recognition, including the development of algorithms that have been widely cited in published papers and operationally implemented. He has made numerous contributions in social network graph analysis as it relates to simulation of social networks, machine learning involving social networks, and construction of networks from multimedia content. Prior to joining Lincoln Laboratory in 2002, he worked on speech processing and communication systems at Motorola. An active contributor to the speech and machine learning research community, he has served as a reviewer and scientific committee member for several conferences: IEEE Odyssey: The Speaker and Language Recognition Workshop; Annual Conference on Neural Information Processing Systems; International Conference on Acoustics, Speech, and Signal Processing; INTERSPEECH; and IEEE Spoken Language Technology Workshop. He is the author of more than 100 peer-reviewed papers, including multiple book chapters; a recipient of the Motorola Distinguished Innovator Award; the holder of 14 patents; and a senior member of the IEEE. He received three bachelor's degrees—in computer science, electrical engineering, and mathematics—from South Dakota School of Mines and Technology in 1990 and master's and doctoral degrees in applied mathematics from Cornell University in 1995.

David J. Weller-Fahy is a technical staff member in the Cyber Analytics and Decision Systems Group and a principal investigator on the CHARIOT program. His research interests include machine learning applications to language problems, programming languages, computer-to-computer communications protocols, and computer and network security. He joined Lincoln Laboratory in 2014. He holds a bachelor's degree in computer science from the University of Illinois Springfield and a master's degree in cyber operations from the Air Force Institute of Technology. For exceptional academic achievement and outstanding research during his graduate studies, he received the Secretary James G. Roche Award.

Alyssa C. Mensch is a technical staff member in the Human Language Technology Group. Since joining Lincoln Laboratory in 2015, she has focused on natural language processing for cyber applications. She received a bachelor's degree in mathematics with computer science from MIT and a master's degree in computer science from the University of Pennsylvania.

Giselle M. Zeno is a doctoral student in the Department of Computer Science at Purdue University. She contributed to the CHARIOT program as a 2015 Lincoln Laboratory Summer Research Program participant in the Human Language Technology Group, applying relational machine learning methods to the cyber domain. At Purdue, she works with Professor Jennifer Neville in the Network Learning and Discovery Lab, studying machine learning in relational domains. She has a bachelor's degree in computer science from the University of Puerto Rico at Bayamón. As a graduate student, she has received two fellowships: one from the National Graduate Degrees for Minorities in Engineering and Science (GEM) Consortium and the other from Purdue University.

Joseph P. Campbell is the associate leader of Lincoln Laboratory's Human Language Technology Group, where he directs the group's research in speech, speaker, language, and dialect recognition; word and topic spotting; speech and audio enhancement; speech coding; text processing; natural language processing; machine translation of speech and text; information retrieval; extraction of entities, links, and events; multimedia recognition techniques, including both voice and face recognition for biometrics applications; and advanced analytics for analyzing social networks on the basis of speech, text, video, and network communications and activities. He specializes in the following for government applications: research, development, evaluation, and transfer of speaker recognition technologies; design of speaker recognition and biometrics evaluations; design of corpora to support those evaluations; and development and evaluation of biometrics technologies and systems. He joined Lincoln Laboratory in 2001 as a senior staff member after serving 22 years at the National Security Agency. He was an IEEE Distinguished Lecturer and is an IEEE Fellow. He earned bachelor's, master's, and doctoral degrees in electrical engineering from Rensselaer Polytechnic Institute in 1979, The Johns Hopkins University in 1986, and Oklahoma State University in 1992, respectively.